Amendments to the Specification:

Please insert the following paragraph before the first paragraph:

This is a Continuation of application 09/107,546, filed 30 June 1998, now USP 6,724,844, issued 20 April 2004.

Please replace the paragraphs beginning at page 4, line 12 with the following rewritten paragraphs:

Figure 1 is a diagram of the standard decision feedback equalizer 10 (DFE). The DFE includes a forward filter 12, a slicer 14, a feedback filter 16 and a subtractor 18. Let a_k be the transmitted trellis-coded symbol stream. The received signal r_k after multi-path distortion and added noise n_k can be written as

$$r_k = \sum_{l=0}^{L_h-1} h_l \, a_{k+d_h-l} + n_k$$

~~after multi-path distortion and added noise n_k can be written as~~

where h_l, i=0... L_h-1 is the multipath channel of length l_h and delay d_h, and n_k is the additive noise which in general is neither gaussian nor white. The forward filter 12 is used to remove the pre echo or ghosts in the received signal. The slicer 14 quantizes the signal ã to the nearest symbol $\hat{a}_k$. If an error is made in this quantization the error is passed to the feedback filter 16 and remains in the system. The output of the feedback filter 16 is subtracted from the output of the forward filter 12 to provide $\tilde{a}_k$. $\tilde{a}_k$ is an estimate of the transmitted symbol a_k plus, an error e_k. It can also be expressed as:

$$a_k = \sum_{l=0}^{L_f-1} f_l \, r_{k+d_f-l} + \sum_{l=1}^{L_b} b_l \, \acute{E}_{k-d_b-l}$$

~~It can also be expressed as:~~

where f_l, i=0... L_f-1 are the forward equalizer taps, b_l, i=1... L_b are the feedback taps, d_f is the delay through the forward equalizer, d_b is the delay in the feedback equalizer and $\hat{a}_k$ is the constellation point closet to $\tilde{a}_k$. In the absence of error propagation, i.e. if $\hat{a}_k = a_k$, the error sequence e_k at the equalizer output 20 is white. However, in most cases the error propagation causes this error sequence to be correlated, that is, the

noise samples are no longer independent. The "colored" noise affects the performance of a trellis coder, because a trellis coder is optimized for performance in a channel having all white guassian noise.

In accordance with the invention, as shown in Fig. 2, an adaptive filter 22 is placed at the output of the DFE 10 but before a trellis decoder (not shown). The output of filter 22 can be expressed as

$$y_k = {}^a{}_k + \sum_{i=1}^{L_g} g_i {}^a{}_{k-i}$$

where $(1, g_1, g_2, g_3 \ldots g_{L_g})$ are the taps of the adaptive filter. Since $\tilde{a}_k = a_k + e_k$, then the adaptive filter output can be written as

$$y_k = a_k + \sum_{i=1}^{L_g} g_i a_{k+i} + e_k + \sum_{i=1}^{L_g} g_i e_{k-i}$$

$$= a_k + \sum_{i=1}^{L_g} g_i a_{k+i} + e'_k$$

~~as~~

If the filter taps are chosen so as to minimize the variance of e'_k, the SNR of the sequence y_k can be improved. In addition, since e'_k is the prediction error sequence of the equalizer output error sequence e_k, it will be white (*see* Widrow and Stearns, "Adaptive Signal Processing," (hereby incorporated by reference), at pages 99-116), which does not affect the performance of a trellis decoder.

Please replace the paragraphs beginning at page 6, line 18 with the following rewritten paragraphs:

To minimize the error from the output of the DFE, or in other words to tune the adaptive filter taps g to the error e_k, the adaptive filter 22 is first placed through a training sequence as shown in Fig. 3. In many applications, such as digital TV, the training sequence is part of the transmitted signal. After the equalizer 10 has converged (via blind means, without using the signal a_k, or trained means which uses the signal a_k) the adaptive filter 22 receives a training sequence which is the

difference between the output of the DFE $\hat{a}_k$ and the transmitted symbol sequence a_k. This difference is the error e_k produced by the DFE.

$$\hat{a} - a_k = e_k \qquad (5)$$

This error sequence e_k is then input into the adaptive filter 22. The adaptive filter 22 forms an output sequence x_k as shown in Fig. 3, where

$$x_k = e_k + \sum_{i=1}^{L_g} g_i e_{k-i}$$

~~3, where~~

The filter taps g_k are adapted by the training device 30 using the LMS algorithm as follows:

$$\underline{g}(k+1) = \underline{g}(k) - \mu x_k \underline{e}(k)$$

~~The filter taps g_k are adapted using the LMS algorithm as follows~~

where $\underline{g}$ = and $\underline{e}(k) = e_{k-1}, e_{k-2}, \ldots e_{k-Lg}]$. This adaptation adapts the filter taps to minimize the mean squared, error e_k^2, of the DFE.

Since the trellis decoder uses the taps $[g_1, g_2, \ldots g_{Lg}]$ in a feedback loop, error propagation can also occur hence it is also beneficial to limit the size of the taps $\underline{g}$ during adaptation, so that the trellis decoder that uses these taps does not suffer error propagation. If g is too small, however, the efficiency of the adaptive filter is reduced. Accordingly, an additional power constraint is imposed on the LMS algorithm to limit the amplitude of the taps which reduces the error propagation in the DDFSE trellis decoder (described next). A parameter P is chosen such that it is required that

$$\sum_{i=1}^{L_g} g_i^2 \le P$$

~~that it is required that~~

At each step of the LMS algorithm this condition is tested, and if violated, the taps are rescaled appropriately.

Please replace the paragraphs beginning at page 8, line 11 with the following rewritten paragraphs:

Once the adaptive filter 22 is appropriately trained to minimize e_k, $\tilde{a}_k = a_k + e_k$ is input to the adaptive filter 22. The adaptive filter 22 although it has been trained to minimize e_k, it will also distort $\tilde{a}_k$. Equation 9 shows this distortion and represents the output of adaptive filter 22 as follows:

$$y_k = {}^{a}{}_k + \sum_{l=1}^{L_g} g_l {}^{a}{}_{k-l} \qquad \text{(9)}$$

~~represents the output of adaptive filter 22 as follows:~~

As stated above, if the filter taps g are chosen to minimize e_k, the SNR of the sequence y_k can be improved. y_k, however, is a distorted $\tilde{a}_k$ and includes the ISI introduced into the data stream by the adaptive filter 22. This distortion is then compensated for in the trellis decoder by use of delayed decision feedback sequence estimation (DDFSE)as described in A. Duel-Hallen and C. Heegard, "Delayed decision-feedback Sequence Estimation," IEEE Trans. Common., Vol. Com - 37, no. 5, pp. 428-436, May 1989, hereby incorporated by reference, and as summarized below.

From the definition of y_k, we see that an optimum trellis decoder which will remove the distortion of $\tilde{a}_k$, should minimize the metric (maximum likelihood decoding):

$$\sum_k \left[y_k - a_k - \sum_{l=1}^{Lg} g_l a_{k-l} \right]^2$$

~~metric (maximum likelihood decoding):~~

This equation, however, requires past symbol sequences a_{k-l} to be saved, and each tap g_l in the adaptive filter therefore introduces more memory into the system which causes the number of steps of a trellis decoder to grow exponentially with the number of taps in the adaptive filter 22. Therefore an expanded trellis is necessary to accommodate the memory (a_{k-l}) introduced by the adaptive filter 22.

A suboptimum, but computationally less intensive way of performing the trellis decoding is to instead minimize the following metric:

$$\sum_{k}\left[y_k - a_k - \sum_{i=1}^{Lg} g_i \acute{E}_{k-i,\,j} \right]^2$$

~~following metric:~~

where the sequence $\hat{a}_{k-i,j}$, i=1, ... L_g is the survivor symbol sequence associated with state j in the trellis. This scheme does not expand the number of states in the original trellis, but instead introduces decision-feedback in each of the trellis states. That is, since this scheme uses $\hat{a}_{k-i}$, which is merely an estimate of, rather than the actual value a_{k-i}, there is no memory introduced into the system. Since there is no memory, an expanded trellis is not required, i.e. the number of states in the trellis remains the same even with the additional adaptive filter 22.